International Research Journal of Engineering and Technology (IRJET) e-ISSN: 2395-0056
Volume: 05 Issue: 03 | Mar-2018 www.irjet.net p-ISSN: 2395-0072

A study paper on Homomorphic encryption in cloud computing

Nivedita W. Wasankar[1], A.V. Deorankar[2]

[1]M. Tech. Scholar,Department of Computer Science and Engineering,Government College of Engineering, Amravati (MH) India
[2]Head of Department,Department of Information Technology,Government College of Engineering, Amravati (MH) India

---

Abstract -*The use of cloud for outsource the database has increased rapidly in many organizations. it provides many benefits in terms of low cost and accessibility of data. Database is hosted and processed in cloud server, which is beyond the control of data owners. due to the privacy concerns that the cloud service provider is assumed semi-trust (honest-but curious.), it becomes a critical issue to put sensitive service into the cloud, so encryption or obfuscation are needed before outsourcing sensitive data. Increased number of queries will inevitably leak more information to the cloud server. One straightforward approach to mitigate the security risk of privacy leakage is to encrypt the private data and hide the query/access patterns. Homomorphic Encryption (HE), a special kind of encryption scheme, can address these concerns as it allows any third party to operate on the encrypted data without decrypting it in advance. This survey focuses on HE and FHE schemes. First,we present the basics of HE and the details of the well-known Partially Homomorphic Encryption (PHE) and Somewhat Homomorphic Encryption (SWHE), which are important pillars of achieving FHE.*

***Key Words*: Homomorfic encryption, FHE, PHE, SWHE, etc..**

1. INTRODUCTION

When the data transferred to the Cloud we use standard encryption methods to secure the operations and the storage of the data. Our basic concept was to encrypt the data before send it to the Cloud provider. But the last one needs to decrypt data at every operation. The client will need to provide the private key to the server (Cloud provider) to decrypt data before execute the calculations required, which might affect the confidentiality and privacy of data stored in the Cloud. One promising direction to preserve the privacy of the data is to utilize homomorphic encryption (HE) schemes. Homomorphic Encryption systems are used to perform operations on encrypted data without knowing the private key (without decryption), the client is the only holder of the secret key. When we decrypt the result of any operation, it is the same as if we had carried out the calculation on the raw data. homomorphic encryption is useful that allows the operations on the cipher text, which can provide the same results after calculations as the working directly on the raw data.

The definition of homomorphic encryption are as follow :

- homomorphism - a transformation of one set into another that preserves in the second set the relations between elements of the first1
- homomorphic encryption - an operation performed on a set of ciphertexts such that decrypting the result of the operation is the same as the result of some operation performed on the plaintexts.

2. POPERTIES OF HOMOMORPHIC ENCRYPTION

An encryption is homomorphic, if: from *Enc(a)* and *Enc(b)* it is possible to compute *Enc(f (a, b))*, where *f* can be: *+, ×,* and without using the private key. Homomorphic Encryption has mainly two properties, according to the operations that allows to assess on raw data.

2.1 Additive Homomorphic Encryption:

A Homomorphic encryption is additive,if

$$Ek(x \; y) = Ek(x) \; Ek(y).$$

2.2Multiplicative Homomorphic Encryption:

Homomorphic encryption is multiplicative, if

$$Ek(x \otimes y) = Ek(x) \otimes Ek(y).$$

- *Ek* is an encryption algorithm with key *k*. - *Dk* is a decryption algorithm.

The additive Homomorphic encryption (only additions of the raw data) is the Pailler and Goldwasser-Micalli cryptosystems, and the multiplicative Homomorphic encryption (only products on raw data) is the RSA and El Gamal cryptosystems. An algorithm is fully homomorphic if both properties are satisfied simultaneously.

3. TYPES OF HOMOMORPHIC ENCRYPTION

Homomorphism is a transformation of one set into another that preserves in the second set the relations between elements of the first one. An operation performed on a set of ciphertexts such that decrypting the result of the operation is the same as the result of some operation performed on the

plaintexts is called as homomorphic encryption. all the different HE schemes can neatly be categorized under three types with respect to the number of allowed operations on the encrypted data as follows:

3.1 Partially Homomorphic Encryption (PHE):

PHE allows only one type of operation with an unlimited number of times (i.e.,no bound on the number of usages). In other words, PHE schemes can only be used for particular applications, whose algorithms include only addition or multiplication operation. PHE schemes are deployed in some applications like e-voting or Private Information Retrieval (PIR). However, these applications were restricted in terms of the types of homomorphic evaluation operations

3.2 Somewhat Homomorphic Encryption (SWHE):

This allows some types of operations with a limited number of times. SWHE schemes support both addition and multiplication. Nonetheless, in SWHE schemes that are proposed before the first FHE scheme, the size of the ciphertexts grows with each homomorphic operation and hence the maximum number of allowed homomorphic operations is limited.

These issues put a limit on the use of PHE and SWHE schemes in real-life applications. Eventually, the increasing popularity of cloud based services accelerated the design of HE schemes which can support an arbitrary number of homomorphic operations with random functions, i.e. FHE.

3.3 Fully Homomorphic Encryption (FHE):

FHE allows an unlimited number of operations with unlimited number of times. The first plausible and achievable Fully Homomorphic Encryption (FHE) scheme was introduced by Craig Gentry in 2009, that evaluates an arbitrary number of additions and multiplications and thus calculate any type of function on encrypted data. It is based on ideal-lattices in math and it is not only a description of the scheme, but also a powerful framework for achieving FHE. However, it is conceptually and practically not a realistic scheme. Different FHE schemes demonstrated that FHE still needs to be improved significantly to be practical on every platform as they are very expensive for real-life applications because of the bootstrapping part, which is the intermediate refreshing procedure of a processed ciphertext.

4. BENEFITS

Homomorphic encryption has many benefits and applications. One such benefit is that of enhanced privacy. Privacy is one of the goals of cryptography in general, but homomorphic encryption can provide even further privacy than typical encryption schemes. One of the biggest benefits to this application is that if a user lives in an area where privacy is considered a luxury, sensitive data can still be retrieved without ever revealing even the nature of the data.

5. DRAWBACKS

While the benefits to homomorphic encryption are great, they do not come without considerable drawbacks. One of the biggest drawbacks is the complexity of the systems. In partially homomorphic cryptosystems, there is not much overhead involved in performing the computations, at least for those presented. However, fully homomorphic encryption requires a lattice-based cryptosystem that is significantly more complex. Implementation of such a cryptosystem even for basic operations requires significantly more complicated computations and massive ciphertext sizes. Another potential drawback of homomorphic cryptosystems is that in some cases, they are vulnerable to malware.

6. CONCLUSION

Homomorphic cryptosystems allow for the same level of privacy as any other cryptosystem, while also allowing for operations to be performed on the data without the need to see the actual data. Indeed, the idea of HE has been around for over 30 years; however, the first plausible and achievable Fully Homomorphic Encryption (FHE) scheme was introduced by Craig Gentry in 2009. Since then, different FHE schemes demonstrated that FHE still needs to be improved significantly to be practical on every platform as they are very expensive for real-life applications. Hence, in this paper, we surveyed the HE and FHE schemes. Specifically, starting from the basics of HE, the details of the well-known Partially HE (PHE) and Somewhat HE (SWHE), which are important pillars of achieving FHE, were presented.

REFERENCES

1. Nitesh Aggarwal, Cp Gupta, and Iti Sharma. 2014. Fully Homomorphic symmetric scheme without bootstrapping.In Cloud Computing and Internet of Things (CCIOT), 2014 International Conference on.IEEE, 14–17.

2. S Sobitha Ahila and KL Shunmuganathan. 2014. State Of Art in Homomorphic Encryption Schemes. International Journal of Engineering Research and Applications 4, 2 (2014), 37–43.

3. Gentry, C. A fully homomorphic encryption scheme. Doctoral Dissertation, Stanford University, 2009.

4. Gentry, C., Sahai, A. and Waters, B. Homomorphic encryption from learning with errors: conceptually-simpler, asymptotically-faster, attribute-based. In Advances in Cryptology, Proceedings of CRYPTO '13. R. Canetti and J. Garay (Eds.). Springer, Berlin Heidelberg, 2013, 75-92.Mufutau Akinwande. 2009.

Advances in Homomorphic Cryptosystems. J. UCS 15, 3 (2009), 506–522.

5. Zvika Brakerski, Craig Gentry, and Vinod Vaikuntanathan. 2014. (Leveled) Fully Homomorphic Encryption Without Bootstrapping. ACM Trans. Comput. Theory 6, 3, Article 13 (July 2014), 36 pages. DOI:http://dx.doi.org/10.1145/2633600.

6. Zhigang Chen, JianWang, ZengNian Zhang, and Song Xinxia. 2014. A fully homomorphic encryption schemewith better key size. Communications, China 11, 9 (2014), 82–92.

7. Michael Clear and Ciarán McGoldrick. 2015. Multi-identity and multi-key leveled FHE from learning witherrors. In Annual Cryptology Conference. Springer, 630–656.

8. Léo Ducas and Daniele Micciancio. 2014. A Fully Homomorphic Encryption library https://github.com/lducas/FHEW. (2014). Accessed at December, 2015.

9. Junfeng Fan and Frederik Vercauteren. 2012a. Somewhat Practical Fully Homomorphic Encryption. IACR Cryptology ePrint Archive 2012 (2012), 144.

10. Shai Halevi and Victor Shoup. 2013b. An Implementation of homomorphi encryption. https://github.com/shaih/HElib. (2013). Accessed at December, 2015.

11. Hao-Miao Yang, Qi Xia, Xiao-fen Wang, and Dian-hua Tang. 2012. A new somewhat homomorphic encryption scheme over integers. In Computer Distributed Control and Intelligent Environmental Monitoring (CDCIEM), 2012 International Conference on. IEEE, 61–64.

12. Goldwasser, S. and Micali, S. Probabilistic encryption. Journal of Computer and System Sciences 28, 2 (1984), 270-299.

13. ElGamal, T. A public key cryptosystem and a signature scheme based on discrete logarithms. In Advances in Cryptology, Proceedings of CRYPTO '84. G. Blakley and D. Chaum (Eds.). Springer, Berlin Heidelberg, 1985, 1018.

14. Rivest, R., Shamir, A. and Adleman, L. A method for obtaining digital signatures and public-key cryptosystems. Communications of the ACM 21, 2 (1978), 120-126.

15. Boneh, D. The decision Diffie-Hellman problem. In Algorithmic Number Theory, Proceedings of the Third International Symposium (ANTS-III) (Portland, June 21-25). J.Buhler (Ed.). Springer, Berlin Heidelberg, 1998, 4863.

16. Paillier, P. Public-key cryptosystems based on composite degree residuosity classes. In Eurocrypt, 1999.

17. Rivest, R., Adleman, L. and Dertouzos, M. On data banks and privacy homomorphisms. In Foundations of Secure Computation 4, 11 (1978), 169-180.

18. Boneh, D., Goh, E.-J. and Nissim, K. Evaluating 2-DNF formulas on ciphertexts. In Theory of Cryptography, Proceedings of the Second Theory of Cryptography Conference (TCC) (Cambridge, February 10-12). J. Kilian (Ed.). Springer, Berlin Heidelberg, 2005, 325-341.